IN THE CLAIMS:

1. (Currently Amended) A processor designed to operate in a plurality of modes for processing vector and scalar instructions, where a vector instruction identifies a single operation to be performed on a plurality of data elements, and a scalar instruction identifies a single operation to be performed on a single data element, comprising:

one or more register files, each for storing scalar and vector data and address information;

a parallel vector unit coupled to receive data from the one or more register files and comprising a plurality of functional units configurable to operate in a vector operation mode and a scalar operation mode, the parallel vector unit includes means for tightly coupling a plurality of the functional units to perform an operation specified by a current instruction;

wherein under a vector operation mode the parallel vector unit performs, in parallel, a single vector operation on a plurality of data elements, the operations performed on the plurality of data elements each being performed by a different functional unit of the parallel vector unit; and

wherein under a scalar operation mode the parallel vector unit performs a scalar operation on a data element received from the one or more register files in a functional unit within the parallel vector unit.

2. (Currently Amended) The processor of claim 1, wherein the parallel vector unit includes a mode selector which reads a mode identifier associated with each instruction and causes the parallel vector unit to enter vector operation mode or scalar operation mode depending on the value of the mode identifier.

3. (Original) The processor of claim 1, further including a power savings unit which disables functional units not used for processing a given instruction.

4. (Original) The processor of claim 3, wherein the functional units are disabled by disabling their clock input signals.

5. (Original) The processor of claim 3, wherein the functional units are disabled by disabling their supply voltage.

6. (Original) The processor of claim 3, wherein the power savings unit disables unused functional units during scalar operations.

7. (Original) The processor of claim 1, further comprising means for designating one or more functional units necessary for execution of an instruction as preferred slots.

8. (Original) The processor of claim 7, wherein the means for designating includes an issue/branch unit coupled to the parallel vector unit, including means for

reading an instruction and determining what functional units are necessary for performing the instruction, and designating such units as preferred slots.

9. (Original) The processor of claim 7, wherein the means for designating includes a tag associated with each datum stored in the one or more register filed indicating whether the datum is valid or invalid data, and enable tag logic coupled to the one or more register files, wherein each tag, if set, causes the enable tag logic to cause one or more functional units to be active for processing an instruction.

10. (Original) The processor of claim 7, wherein the preferred slots are designated at system design.

Please add new Claims 11-21 as follows:

11. (New) The processor of claim 1, wherein the one or more registers each further store conditional information.

12. (New) The processor of claim 2, wherein each of the one or more registers comprises at least one mode identifier field for storing at least one mode identifier therein.

13. (New) The processor of claim 12, further comprising means for designating one or more functional units necessary for execution of an instruction as preferred slots, and wherein the at least one mode identifier field comprises information indicating whether values in a current slot are valid and an operation is to be performed on a particular slot, or whether the values in the current slot are invalid and associated ones of the plurality of functional units are to be powered down.

14. (New) The processor of claim 12, wherein the at least one mode identifier field comprises a plurality of mode identifier fields, each of the one or more registers comprises a plurality of register segments, and each of the plurality of register segments is associated with a different one of the plurality of mode identifier fields.

15. (New) The processor of claim 14, wherein the at least one mode identifier is dynamically determined based on pre-identified criteria.

16. (New) The processor of claim 15, wherein the pre-identified criteria comprises whether a current operation is widening or narrowing.

17. (New) The processor of claim 15, further comprising means for designating one or more functional units necessary for execution of an instruction as preferred slots, and wherein the mode identifier includes decoding information for a given operation, the decoding information comprising whether the given operation increases, decreases, changes or maintains a number and location of valid slots.

18. (New) The processor of claim 17, wherein the decoding information further comprises information relating to powering down of corresponding ones of the plurality of functional units.

19. (New) The processor of claim 2, wherein, for a given instruction, a corresponding mode identifier is evaluated to determined if computational consistency has been maintained or if an exceptional situation has occurred with respect to the given instruction.

20. (New) The processor of claim 1, wherein each of the one or more registers comprises a plurality of register segments, each of the plurality of register segments being responsive to a selection thereof by a multiplexer.

21. (New) A processor designed to operate in a plurality of modes for processing vector and scalar instructions, where a vector instruction identifies a single operation to be performed on a plurality of data elements, and a scalar instruction identifies a single operation to be performed on a single data element, comprising:

one or more merged Single Instruction Stream, Multiple Data Stream (SIMD)/scalar register files, each for storing both scalar and vector data;

one or more address register files, each for storing address information;

a parallel vector unit coupled to receive data from the one or more merged SIMD/scalar register files and comprising a plurality of functional units configurable to

operate in a vector operation mode and a scalar operation mode, the parallel vector unit includes means for tightly coupling a plurality of the functional units to perform an operation specified by a current instruction;

wherein under a vector operation mode the parallel vector unit performs, in parallel, a single vector operation on a plurality of data elements, the operations performed on the plurality of data elements each being performed by a different functional unit of the parallel vector unit; and

wherein under a scalar operation mode the parallel vector unit performs a scalar operation on a data element received from the one or more merged SIMD/scalar register files in a functional unit within the parallel vector unit.